

**Universidad de Costa Rica  
Facultad de Ingeniería  
Escuela de Ingeniería Eléctrica**

**IE0409 Análisis de Sistemas**

**Brazos Roboticos**

**por**

Jordy Cordero Montero, carné B72306  
Royer Méndez Ramírez, carné A43333  
Jordi Louzao von Breymann, carné B53993

**I-2020**

### **Resumen**

En este trabajo se estudia un sistema mecánico no lineal conformado por un brazo robótico de dos enlaces con dos grados de libertad, tomado de un artículo científico y mediante este se crea un modelo en variables de estado que describa el sistema para luego hacer simulaciones en MatLab y corroborar resultados válidos.

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Alcances . . . . .	2
1.2. Objetivos . . . . .	2
1.2.1. Metodología . . . . .	2
<b>2. Marco teórico</b>	<b>4</b>
2.1. Tipos y clasificación de brazos robóticos: . . . . .	5
2.2. Servo y Non-Servo: . . . . .	6
2.3. Descripción del sistema: . . . . .	6
2.4. Grados de libertad: . . . . .	6
<b>3. Modelado y simulación</b>	<b>8</b>
3.1. Modelo planteado por GUECHI y col. 2018. . . . .	8
3.2. MVE a partir del modelo dado . . . . .	9
3.3. Simulación . . . . .	11
<b>4. Análisis</b>	<b>17</b>
<b>5. Conclusiones y recomendaciones</b>	<b>19</b>
5.1. Conclusiones . . . . .	19
5.2. Recomendaciones . . . . .	19
<b>A. Código de s-function</b>	<b>21</b>

# Capítulo 1

## Introducción

Los brazos robóticos se han convertido en una herramienta funcional muy eficiente y eficaz para realizar tareas pesadas o de alta precisión en la industria, es por esta razón que mediante este estudio se plantea investigar un prototipo de brazo robótico mecánico de dos enlaces.

### 1.1. Alcances

Mediante el análisis del artículo dado, el trabajo plantea un sistema mecánico tipo brazo de dos enlaces en el que se estudia el movimiento del sistema con dos grados de libertad. Tomando el planteamiento físico y analizando el estudio hecho por GUECHI y col. 2018 se observa el modelo matemático planteado y además se plantea un modelo matemático en la forma MVE que se utiliza para hacer simulaciones en MatLab y de esta manera obtener resultados. Además observar que los resultados concuerden satisfactoriamente por medio de algunas pruebas.

### 1.2. Objetivos

Se establecieron los siguientes objetivos para el desarrollo del trabajo:

- Comprender el funcionamiento físico de un brazo robótico de dos enlaces.
- Entender un modelo matemático planteado para un brazo robótico.
- Encontrar un modelo matemático del sistema a partir de su estructura física y el modelo matemático dado.
- Simular resultados a partir del software MatLab y comprobar concordancia.

#### 1.2.1. Metodología

Para el desarrollo del trabajo se pretende incluir los siguientes pasos y procedimientos:

1. Revisión de material bibliográfico para comprender conceptos importantes acerca del tema de estudio.

2. Estudiar un brazo robótico como sistema, ya expuesto en una de las referencias y analizar el modelo matemático planteado de manera que se entienda la dinámica del sistema.
3. Tomando como base el modelo dinámico hecho por GUECHI y col. 2018, modificarlo para convertir a MVE.
4. Mediante el Software MatLab simular el MVE obtenido en el punto anterior y observar el comportamiento obtenido.

## Capítulo 2

# Marco teórico

Con el desarrollo de la tecnología se ha ido evolucionando la técnica para realizar tareas que antes resultaban muy difíciles de realizar o que simplemente eran imposibles de realizar por los seres humanos.

Con el avance en el conocimiento de la electricidad ha sido posible y muy conveniente la creación de sistemas electromecánicos de producción que facilitan la masificación de los productos y aumentan la producción en la industria.

Gracias a este avance se han ido dejando de utilizar otros métodos de trabajo como la combustión (Ej. trenes de combustión interna) o el trabajo manual que eran de mayor costo, ocupaban mucho más espacio, eran inseguros y hasta difíciles de realizar.

Con el perfeccionamiento de la electrónica, se han podido disminuir los sistemas electromecánicos a dimensiones que antes no se hubieran pensado, a la vez que estos se han impregnado de “inteligencia” para realizar funciones.

Ahora no solo son codiciados para la producción en serie en las industrias sino para todo tipo de aplicaciones en la vida diaria, como lo menciona el siguiente párrafo “Este tipo de robots son una herramienta para no depender tanto de otras personas, especialmente cuando se trata de tareas que requieren de manipulación de objetos que pueden provocar lesiones.” (Semanario Universidad 2019)

El departamento de trabajo de los Estados Unidos define los robots industriales como “dispositivos mecánicos multifuncionales programables diseñados para mover material, piezas, herramientas ó dispositivos especializados a través de movimientos programados variables para realizar una variedad de tareas. Un sistema de robot industrial incluye no solo robots industriales, sino también cualquier dispositivo y / o sensor requerido para que el robot realice sus tareas, así como la secuenciación o monitoreo de interfaces de comunicación.”(Departamento de Trabajo de los Estados Unidos 2019)

Los brazos robóticos son fundamentales en la automatización de sistemas debido a que se pueden programar para que cumplan determinados movimientos infinita cantidad de veces, y se pueden reprogramar para cumplir después otra serie de movimientos siempre y cuando esté dentro de su rango de posibilidades físicas y tecnológicas.

## 2.1. Tipos y clasificación de brazos robóticos:

Existe una diversa variedad de brazos robóticos en el mercado. Son fabricados con diferentes diseños de configuración y con diferentes grados de libertad dependiendo de la necesidad del trabajo a realizar.

Se pueden mencionar los siguientes:

- Robot de coordenadas rectangulares.
- Robot de coordenadas cilíndricas.
- Robot de coordenadas esféricas.
- Robot de brazo articulado (este se estudiará en el presente informe).
- Robot SCARA.

La figura 2.1 muestra los diagramas de estos sistemas.

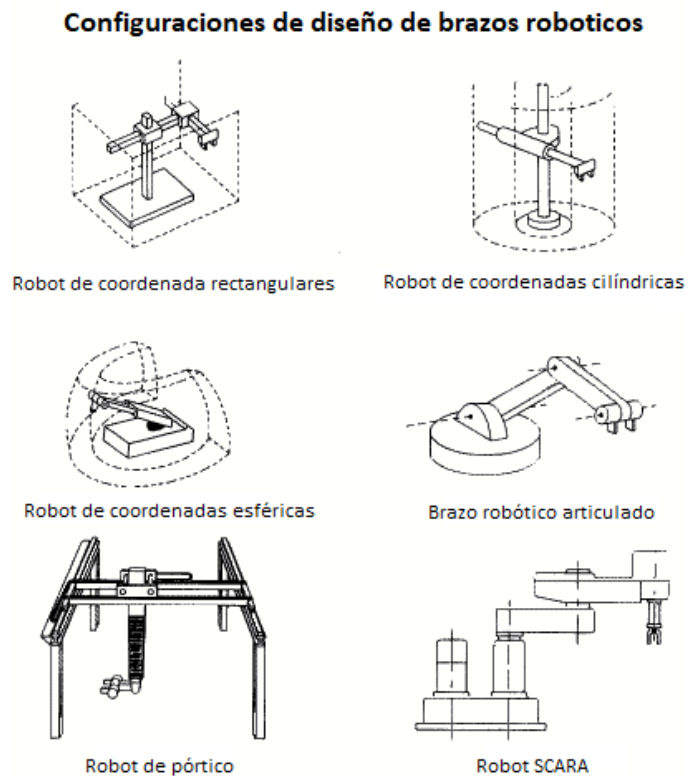


Figura 2.1: Tipos de brazos (Departamento de Trabajo de los Estados Unidos 2019)

## 2.2. Servo y Non-Servo:

Los “robots Servo” son robots con mayor libertad de actuar. Este mecanismo comprende un motor colocado en las articulaciones de los brazos, cuya finalidad es la de controlar esa articulación. Este tipo de mecanismos puede girar y también hacer que el brazo se estire hasta el punto específico que se desea.

Los robots “Non-Servo” son robots que funcionan axialmente; primero se mueven en un eje y después se mueven en el otro eje para lograr su objetivo.

Estos últimos son más fáciles de programar que los primeros, sin embargo, su libertad de acción es más limitada debido a que no pueden llegar a todas las partes como sí lo pueden hacer los brazos tipo servo.

El brazo robótico que se explorará en este proyecto es de tipo servo.

## 2.3. Descripción del sistema:

El sistema del brazo robótico se subdivide en tres partes:

- **Interfaz gráfica:** En esta etapa se elaborará el modelo del sistema con base en los parámetros requeridos por medio de un ordenador con Matlab u otro programa afín. La información que se obtenga después alimentará al controlador. (Sivraj 2017)
- **Controlador:** Arduino, Raspberry Pi u otro controlador afín. Encargado de recibir la información del ordenador, transformarla en señales y enviarla a los motores del brazo. (Sivraj 2017)
- **Brazo:** Este es todo el aparato físico que realizará el trabajo. Está compuesto por la unión del hombro, los brazos, el codo, la muñeca. Cada articulación utiliza un motor. (Sivraj 2017)

## 2.4. Grados de libertad:

Dependiendo del brazo robótico que se manipule, existirán ciertas direcciones hacia las cuales el podrá moverse. En el espacio tridimensional, el brazo puede tener tres direcciones de movimiento: arriba-abajo, izquierda-derecha, adentro-afuera, pero además puede tener también movimientos de muñeca adicionales en los extremos de cada brazo que son: de arriba a abajo, de izquierda a derecha y rotacional (sentido horario y anti-horario).

El actual trabajo emplea un brazo robótico de dos grados de libertad.

La figura 2.2 muestra el espacio de trabajo de un brazo robótico de tres grados de libertad a manera de ejemplo.



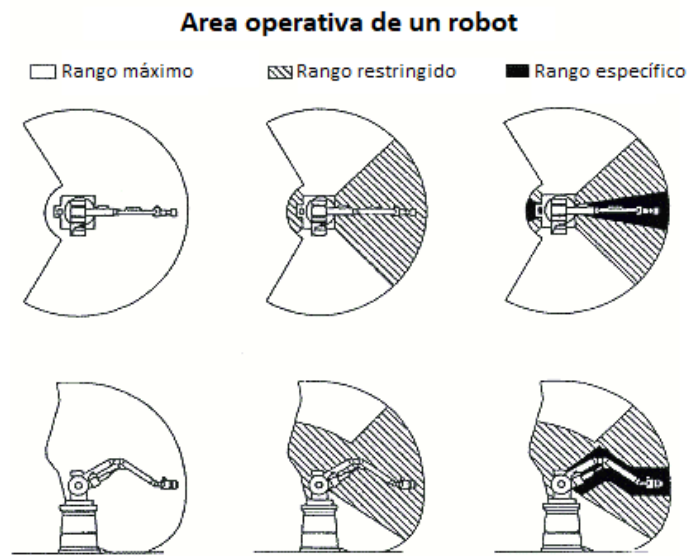


Figura 2.2: Grados de libertad (Departamento de Trabajo de los Estados Unidos 2019)

## Capítulo 3

# Modelado y simulación

### 3.1. Modelo planteado por GUECHI y col. 2018.

El modelo a utilizar es un modelo matemático tal como fue planteado en el trabajo "Model predictive control of a two-link robot arm"(GUECHI y col. 2018). En este trabajo se plantea un modelo matemático para un brazo robótico con dos grados de libertad. El mismo corresponde a un sistema no lineal, que se analizará sin aplicar un método de linealización. Primero se definen los parámetros del sistema. Estos son la longitud y masa de cada segmento del brazo. Además de esto se definen los ángulos entre el primer segmento de brazo y el eje y,  $\theta_1$ , y el ángulo del segundo segmento respecto al primer segmento,  $\theta_2$ . Se definen  $M_1$  y  $M_2$  como las masas de cada segmento del brazo, y  $L_1$  y  $L_2$  como las longitudes de cada segmento del brazo. El sistema que se propone en el trabajo se basa en energías cinéticas y potenciales. Se utilizan las funciones que modelan la posición x y y del sistema como un punto de partida para implementar el modelo usando energía.

$$x = L_1 \sin(\theta_1) + L_2 \sin(\theta_2) \quad (3.1)$$

$$y = L_1 \cos(\theta_1) + L_2 \cos(\theta_2) \quad (3.2)$$

A partir de esto, definen las ecuaciones para la energía cinética y la energía potencial.

$$E = \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2L_2^2\dot{\theta}_1^2 + M_2L_2^2\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}M_2L_2^2\dot{\theta}_2^2 + M_2L_1L_2(\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2)\cos(\theta_2) \quad (3.3)$$

$$U = M_1gL_1 \cos(\theta_1) + M_2g(L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)) \quad (3.4)$$

Utilizan el formalismo de Lagrange para determinar las ecuaciones de movimiento del brazo.

$$L = E - U \quad (3.5)$$

Con L resuelven la ecuación de Euler-Lagrange que utiliza la derivada parcial de las energías cinéticas y potenciales para determinar las funciones de movimiento del brazo.

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} \quad (3.6)$$

En esta ecuación  $L$  y  $\tau$  corresponden respectivamente al Lagrangiano del vector de movimiento y torque. Finalmente a partir de esto crean el modelo dinámico del brazo robótico con dos grados de libertad.

$$\begin{cases} M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = \tau \\ Y = \theta \end{cases} \quad (3.7)$$

con

- $\theta = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T$  es el vector de variables de las articulaciones
- $\tau = \begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix}^T$  es el vector de torque, que se refiere a las entradas de control
- $Y$  es el vector de salidas
- $G(\theta) = \begin{bmatrix} -(M_1 + M_2)gL_1 \sin(\theta_1) - M_2gL_2 \sin(\theta_1 + \theta_2) \\ -M_2gL_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$  es el vector de torques debido a gravedad
- $C(\theta, \dot{\theta}) = \begin{bmatrix} -M_2L_1L_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2) \sin(\theta_2) \\ -M_2L_1L_2\dot{\theta}_1\dot{\theta}_2 \sin(\theta_2) \end{bmatrix}$  es el vector de Coriolis y fuerzas centrípetas
- $M(\theta) = \begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix}$  es la matriz de inercia donde:
  - $D_1 = (M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2 \cos(\theta_2)$
  - $D_2 = M_2L_2^2 + M_2L_1L_2 \cos(\theta_2)$
  - $D_3 = D_2$
  - $D_4 = M_2L_2^2$

Este es el modelo del sistema dinámico presentado en GUECHI y col. 2018. En el trabajo se linealiza diferenciando la salida  $Y$  hasta que la entrada de control  $\tau$  aparezca. Con esto se obtiene la ley de control

$$\tau = M(\theta)v + C(\theta, \dot{\theta}) + G(\theta) \quad (3.8)$$

Se combinan 3.7 con 3.8 para obtener las ecuaciones del sistema linealizadas. En este caso, obtienen un sistema lineal para cada articulación, en términos de la variable de Laplace  $p$ .

$$\frac{\theta_1(p)}{v_1(p)} = \frac{1}{p^2} \quad y \quad \frac{\theta_2(p)}{v_2(p)} = \frac{1}{p^2} \quad (3.9)$$

### 3.2. MVE a partir del modelo dado

Como se observa en la sección 3.1 el modelo planteado llega a un resultado en el que se necesita una etapa realimentación para observar los resultados y es por ello que se crea la etapa de control del brazo, sin embargo para el análisis lo que se quiere es obtener es un MVE y observar la respuesta del sistema, es decir, del modelo matemático planteado en la sección 3.1 se quiere modificar de tal manera que se obtenga la respuesta al sistema sin la etapa de control.

Para ello lo primero es expandir la ecuación 3.7, como es una ecuación matricial lo que se forman son dos ecuaciones, es decir tenemos un sistema de ecuaciones diferenciales no lineales de orden 2.

Luego haciendo álgebra en el sistema de ecuaciones tenemos:

$$\begin{cases} \ddot{\theta}_1 + \alpha_1(\theta_1, \theta_2)\ddot{\theta}_2 = f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \\ \ddot{\theta}_2 + \alpha_2(\theta_1, \theta_2)\ddot{\theta}_1 = f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \end{cases} \quad (3.10)$$

donde:

- $\alpha_1(\theta_1, \theta_2) = \frac{M_2 L_2^2 + M_2 L_1 L_2 \cos(\theta_2)}{(M_1 + M_2)L_1^2 + M_2 L_2^2 + 2M_2 L_1 L_2 \cos(\theta_2)}$
- $\alpha_2(\theta_1, \theta_2) = \frac{M_2 L_2^2 + M_2 L_1 L_2 \cos(\theta_2)}{M_2 L_2^2}$
- $f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \frac{\tau_1 + L_1 L_2 M_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2) \sin(\theta_2) + g(M_1 + M_2)L_1 \sin(\theta_1) + gM_2 L_2 \sin(\theta_1 + \theta_2)}{L_1^2(M_1 + M_2) + L_2^2 M_2 + 2M_2 L_1 L_2 \cos(\theta_2)}$
- $f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \frac{\tau_2 + L_1 L_2 M_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_2) + g(M_1 + M_2)L_1 \sin(\theta_1) + gM_2 L_2 \sin(\theta_1 + \theta_2)}{M_2 L_2^2}$

Luego  $f_1$  y  $f_2$  se pueden reescribir de la siguiente forma:

$$A \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{pmatrix} 1 & \alpha_1 \\ \alpha_2 & 1 \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

Se quiere despejar las variables  $\ddot{\theta}_1$  y  $\ddot{\theta}_2$  entonces se multiplica a ambos lados por la matriz inversa de A sin embargo A es invertible siempre que  $\text{Det}(A) > 0$ , entonces se calcula el determinante y se prueba que para valores limite se cumpla la condición.

$$\text{Det}(A) = 1 - \alpha_1 \alpha_2 = 1 - \frac{(1 + \cos(\theta_2))^2}{3 + 2\cos(\theta_2)}$$

- Cuando  $\cos(\theta_2) = 0 \rightarrow \text{Det}(A) = 0,667 > 0$
- Cuando  $\cos(\theta_2) = 1 \rightarrow \text{Det}(A) = 0,2 > 0$
- Cuando  $\cos(\theta_2) = -1 \rightarrow \text{Det}(A) = 1 > 0$

Por lo tanto podemos invertir A y utilizar el método antes indicado quedando la expresión de  $\ddot{\theta}_1$  y  $\ddot{\theta}_2$  de la siguiente manera:

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \frac{f_1 - \alpha_1 f_2}{1 - \alpha_1 \alpha_2} \\ \frac{-\alpha_2 f_1 + f_2}{1 - \alpha_1 \alpha_2} \end{bmatrix} \quad (3.11)$$

De esta manera despejamos las variables  $\ddot{\theta}_1$  y  $\ddot{\theta}_2$  que representan a  $\dot{\omega}_1$  y  $\dot{\omega}_2$  respectivamente que son variables de estado derivados.

Por lo tanto el MVE del brazo robótico es:

$$\begin{cases} \dot{\theta}_1 = \omega_1 \\ \dot{\theta}_2 = \omega_2 \\ \dot{\omega}_1 = \frac{f_1 - \alpha_1 f_2}{1 - \alpha_1 \alpha_2} \\ \dot{\omega}_2 = \frac{-\alpha_2 f_1 + f_2}{1 - \alpha_1 \alpha_2} \end{cases} \quad (3.12)$$

### 3.3. Simulación

Para realizar la simulación, se utilizó la s-function de MATLAB para generar un bloque de Simulink. Este bloque generaría las salidas del sistema, que en este caso serían los ángulos  $\theta_1$  y  $\theta_2$  correspondientes a cada sección del brazo. El sistema recibe de entradas los torques  $\tau_1$  y  $\tau_2$ , y en sus parámetros se recibe un vector de condiciones iniciales para los estados, y un segundo vector que contiene los parámetros del sistema  $M_1$ ,  $M_2$ ,  $L_1$ , y  $L_2$ . De esta manera, se pueden variar todos los factores de interés para el análisis del sistema.

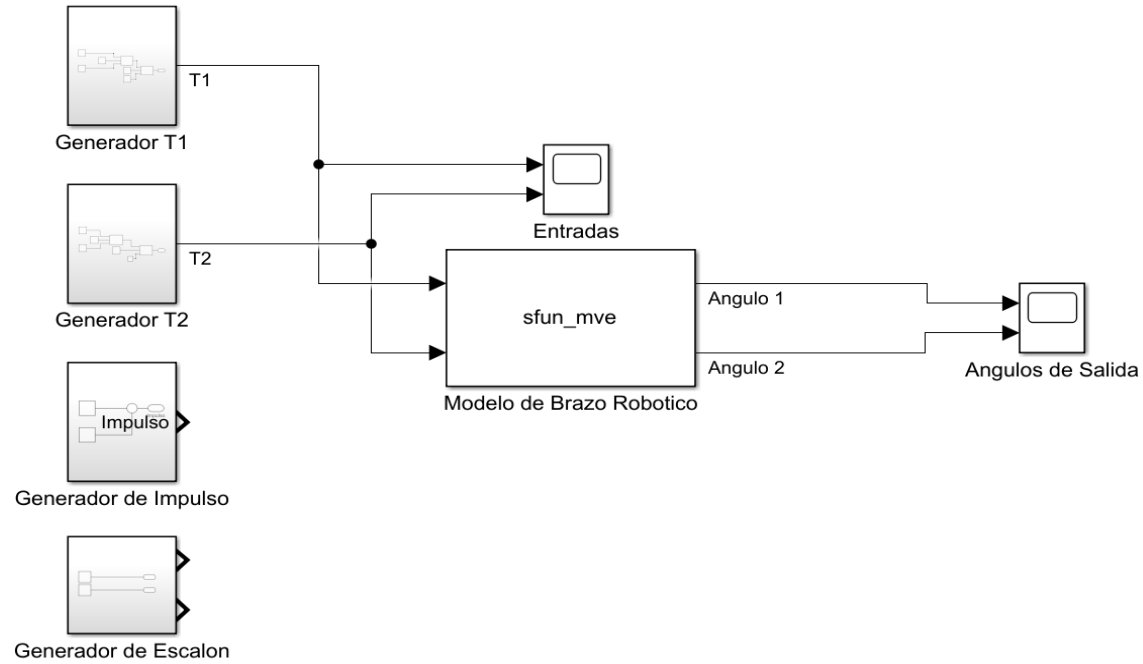


Figura 3.1: Modelo de Simulink

Como se puede notar en 3.1, se crearon varios subsistemas que generan una variedad de señales para los torques de entrada al sistema. En el caso de los generadores T1 y T2 mostrados, estos intentan replicar la señal expuesta en el trabajo de donde se obtuvo el modelo.(GUECHI y col. 2018) Para lograr esto se utilizaron bloques sumadores, interruptores, y generadores de señales senoidales. También se generan señales de impulso y de escalón unitario para realizar pruebas adicionales. Para visualizar los resultados de la simulación se utilizan dos bloques para ver las señales de entrada y salida del sistema. Para las pruebas se utilizan valores de  $M_1 = M_2 = 1$  kg,  $L_1 = L_2 = 1$  m, a menos que se especifique otro valor.

Para la primera prueba, se obtiene la respuesta del sistema ante una entrada cero, y condiciones iniciales de  $\theta_1 = \frac{\pi}{2}$   $\theta_2 = -\frac{\pi}{2}$ , tal como se plantea en el trabajo en que se obtuvo el modelo. El propósito de esta prueba es analizar la respuesta natural del sistema, sin que se tomen en cuenta las entradas.

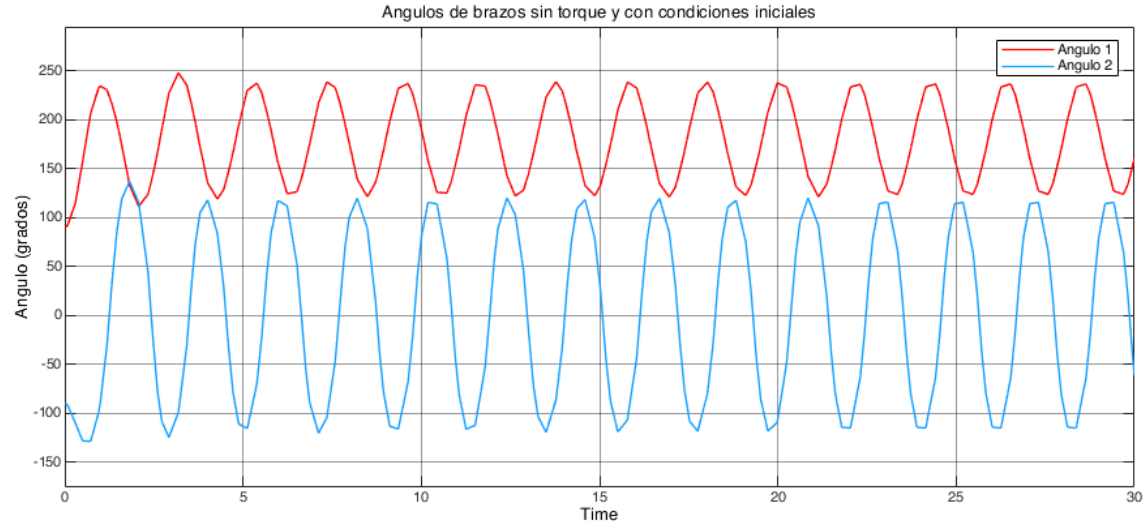


Figura 3.2: Respuesta ante condiciones iniciales y sin entradas

La aproximación al los torques de control utilizados por GUECHI y col. 2018 se genera en un subsistema del modelo de Simulink, y se obtiene esta señal.

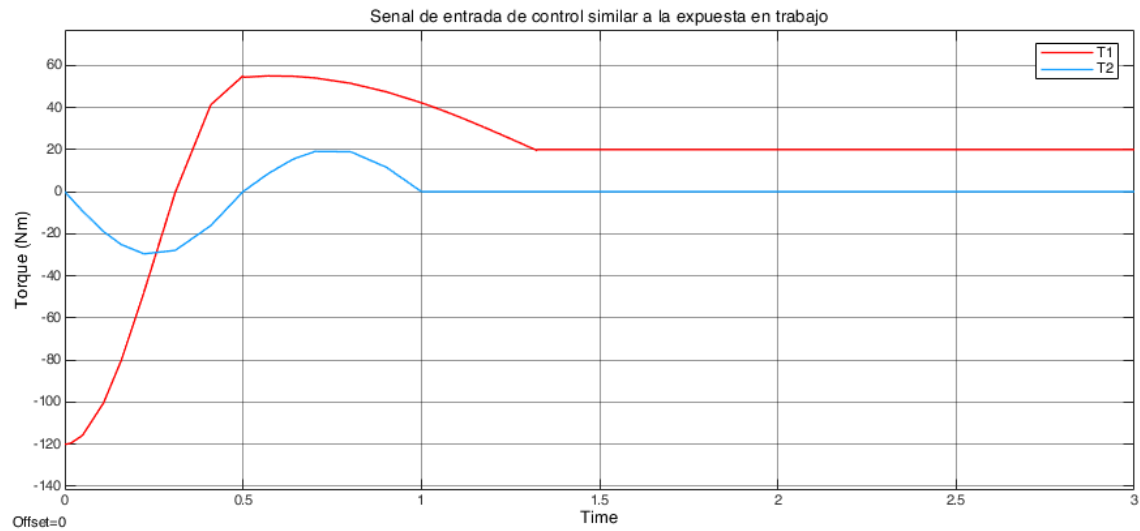


Figura 3.3: Torques de entrada a utilizar

Entonces, ante esta entrada se simula la respuesta del sistema con condiciones iniciales cero. El propósito de esta prueba es ver como esta aproximación al torque de control afecta el sistema sin la influencia de otros factores.



Figura 3.4: Respuesta ante condiciones iniciales cero y torques de entrada

También se realiza la misma prueba pero ahora con las condiciones iniciales especificadas por GUECHI y col. 2018.

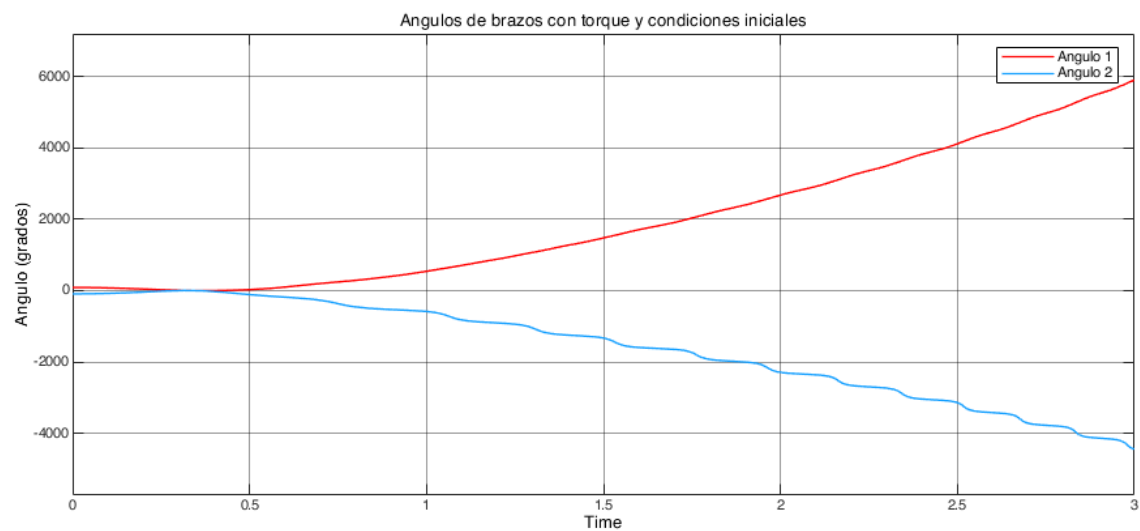


Figura 3.5: Respuesta ante condiciones iniciales y torque de entrada

Se realizan pruebas variando únicamente los parámetros de longitud de los segmentos del brazo. En este caso se ocupan definir condiciones iniciales distintas a cero, ya que el sistema simplemente daría una respuesta cero ante una entrada cero y condiciones iniciales cero.

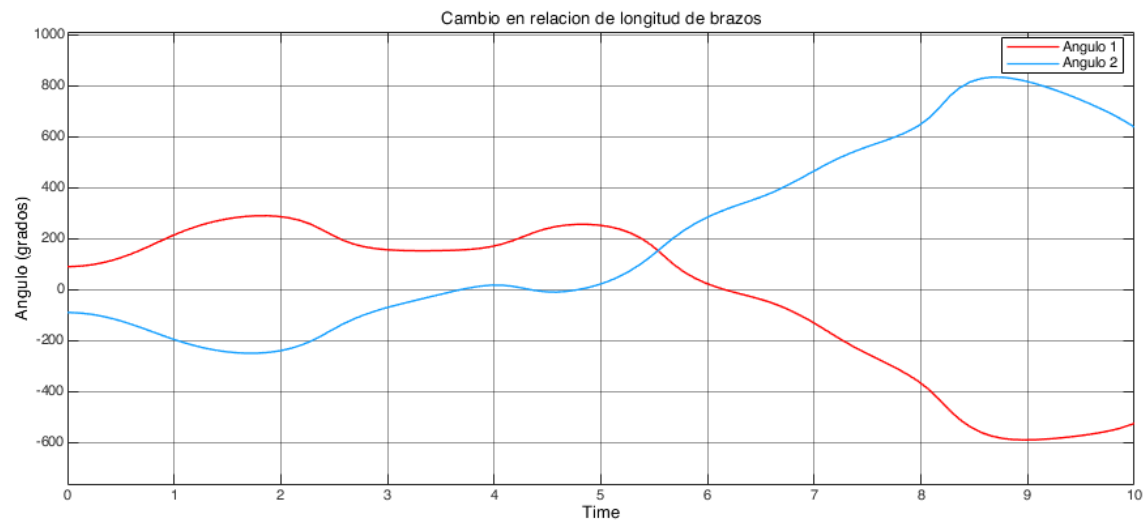


Figura 3.6: Respuesta ante condiciones iniciales sin entradas y parámetros  $L1 = 2m$   $L2 = 10m$

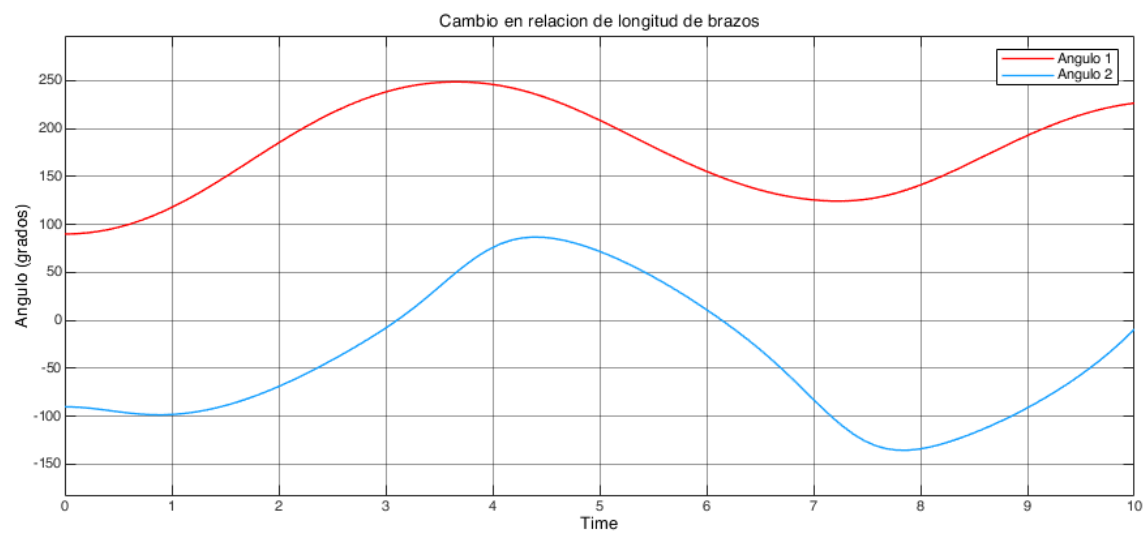


Figura 3.7: Respuesta ante condiciones iniciales sin entradas y parámetros  $L1 = 10m$   $L2 = 2m$



Ahora se simula la respuesta del sistema ante una entrada de torque correspondiente a un escalón unitario en cada una de sus entradas. Esta prueba nos permite ver como reacciona el sistema ante una entrada constante.

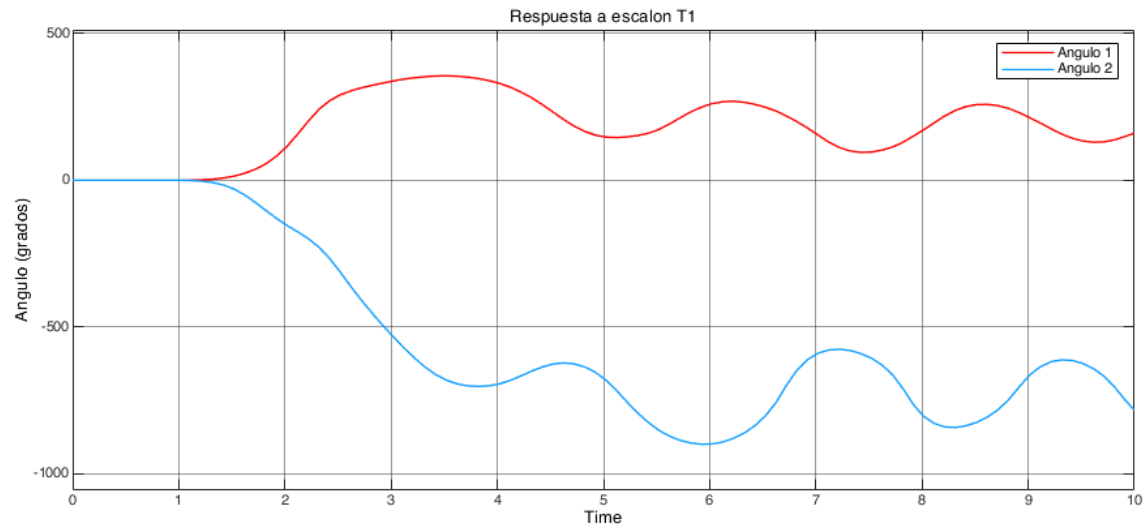


Figura 3.8: Respuesta ante escalón unitario en  $\tau_1$

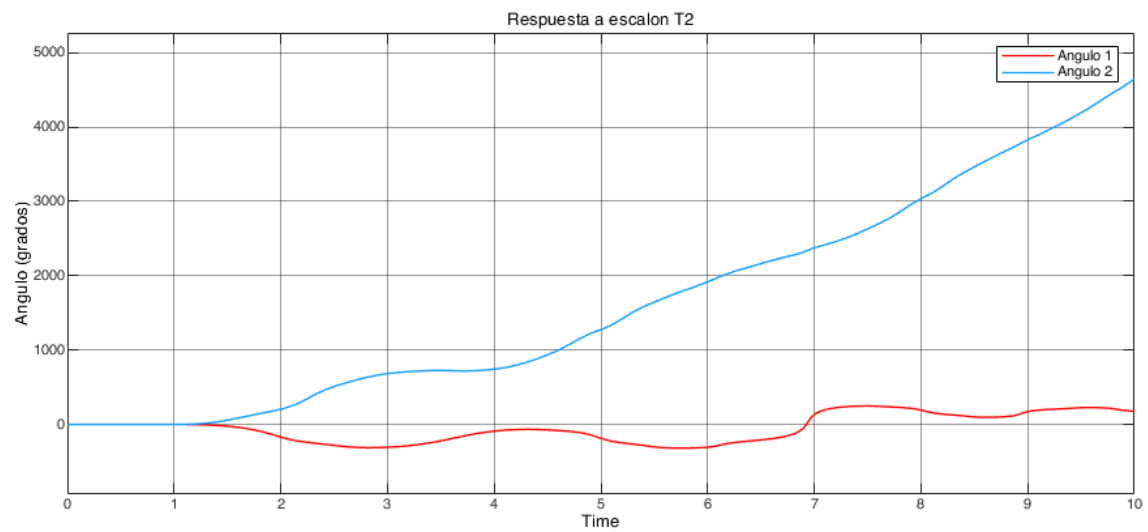


Figura 3.9: Respuesta ante escalón unitario en  $\tau_2$

También se simula la respuesta del sistema ante un impulso unitario con una duración de 0.25s en cada una de las entradas. El propósito de esta prueba sería identificar como se degrada el sistema después de un impulso inicial. Esto nos permite ver como las salidas del sistema cambian, si se estabilizan, o si oscilan.

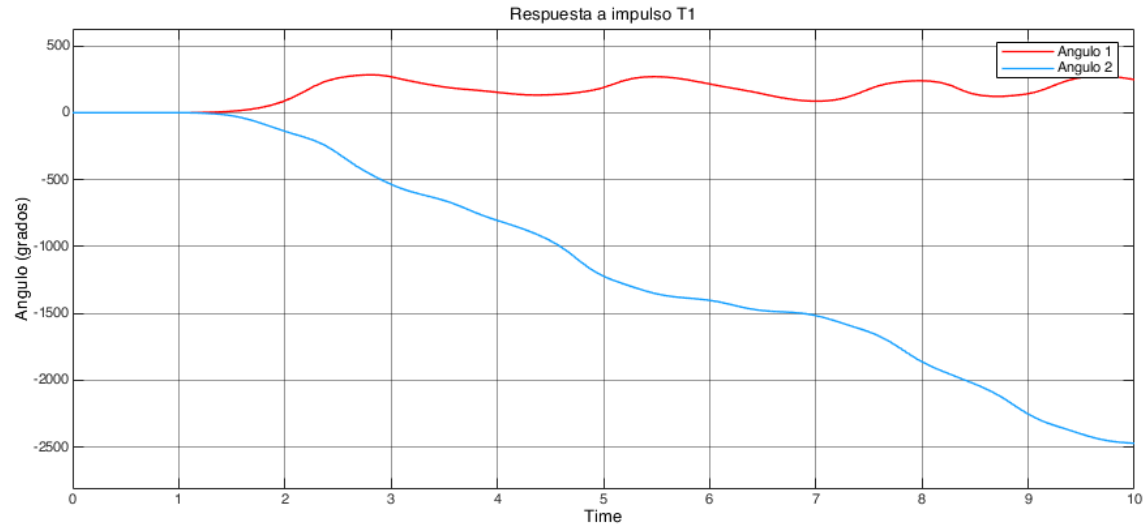


Figura 3.10: Respuesta ante impulso unitario en  $\tau_1$

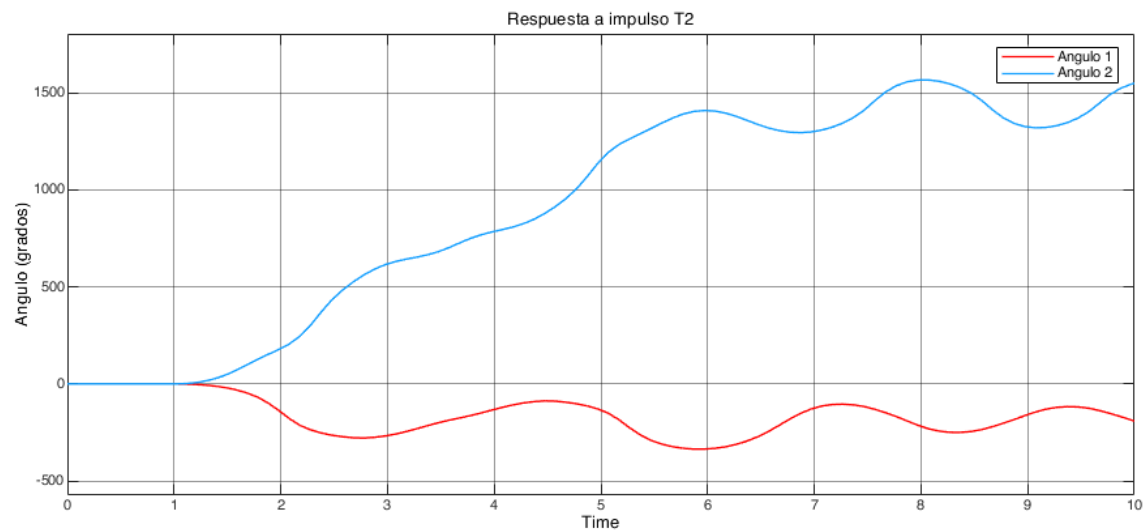


Figura 3.11: Respuesta ante impulso unitario en  $\tau_2$

## Capítulo 4

# Análisis

Es importante recalcar antes de iniciar el comentario de los resultados que el modelo del sistema a utilizar no es lineal. Dado que el modelo planteado por el trabajo de GUECHI y col. 2018 lo linealizan y controlan a partir de esto, se puede esperar que el modelo simulado aquí tenga respuestas diferentes. Dicho esto, primero se analiza la respuesta obtenida en la figura 3.2, correspondiente a una entrada nula con condiciones iniciales de  $\theta_1 = \frac{\pi}{2}$  y  $\theta_2 = \frac{-\pi}{2}$ . Se nota que en ambas salidas se presenta una respuesta oscilatoria no amortiguada. Dado que el brazo sin ningún torque de entrada actuaría básicamente como un péndulo doble, esta respuesta es la esperada. La falta de amortiguamiento también es consistente con lo esperado, ya que el modelo toma en cuenta únicamente fuerzas gravitacionales, centrípetas, y de Coriolis, y no toma en cuenta factores como la fricción en las articulaciones del brazo, o la fricción con el aire que causarían que el movimiento del brazo fuera amortiguado.

Para la siguiente prueba, se definen condiciones iniciales cero y se aplica una señal de torque similar a la calculada por GUECHI y col. 2018 en su trabajo, que se puede observar en la figura 3.3. En la figura 3.4 se observan los resultados de esta prueba. Se nota que los ángulos del sistema crecen de manera no acotada, y se puede observar en ambos una leve oscilación. En términos concretos, esto correspondería a que el primer segmento del brazo rotaría constantemente en sentido horario, y que el segundo segmento haría lo mismo en sentido contra-horario. Dado que el modelo no considera pérdidas por fricción, es lógico que una entrada de torque constante genere una respuesta no acotada. Se está haciendo trabajo sobre el sistema, y el modelo no considera pérdidas de energía, entonces naturalmente este continuara creciendo su salida de manera no acotada.

En la siguiente prueba, se introducen las condiciones iniciales  $\theta_1 = \frac{\pi}{2}$  y  $\theta_2 = \frac{-\pi}{2}$  y además se introduce la misma señal de torque que en el punto anterior. Observando las salidas que se grafican en la figura 3.5 se nota una respuesta muy similar a la anterior. Un factor importante es la reducción en las oscilaciones en las salidas del sistema. En el caso de  $\theta_1$  se nota una línea prácticamente sin oscilaciones, mientras que en el caso de  $\theta_2$  se nota una reducción importante en la magnitud de las oscilaciones. Lo que esto nos indica es que al utilizar una señal de control similar a la planteada en el trabajo referenciado (GUECHI y col. 2018), con las mismas condiciones iniciales y parámetros, se obtuvo una salida con mucha menos variación. A partir de esto podríamos decir que utilizar estas entradas y condiciones iniciales logra de cierta manera estabilizar el sistema, ya que se atenúa su naturaleza oscilatoria y produce una salida que cambia de manera predecible.

En la figura 3.6 y 3.7 se observa la respuesta del sistema con entradas cero, condiciones iniciales  $\theta_1 = \frac{\pi}{2}$  y  $\theta_2 = \frac{-\pi}{2}$ , pero variando la relación de longitud de los brazos. En un caso se definen  $L_1 = 2$  m y  $L_2 = 10$  m. En este caso se observa como la relación entre los ángulos es parecida en la forma general, pero reflejada

como en un espejo. Recordando que ante una entrada nula el sistema se comporta como un péndulo doble, se sabe que el primer segmento del brazo tiene una masa igual que el segundo, pero concentrada en una longitud mucho menor. Esto implica que el par que este segmento aplica sobre el segundo segmento es mucho menor que el par que genera el segundo segmento sobre el primer segmento. En el caso opuesto, donde  $L_1 = 10$  m y  $L_2 = 2$  m, se nota que los ángulos tienen una curva muy similar, pero desplazada hacia abajo. En este caso, el par que aplica el primer segmento de brazo es mucho mayor al que aplica el segundo segmento. Esto resulta en que el primer segmento defina básicamente la trayectoria a seguir, y que el segundo segmento simplemente seguirá esta trayectoria, manteniendo el desfase original dado por las condiciones iniciales.

En las figuras 3.8 y 3.9 se muestran las respuestas ante una entrada de escalón unitario para cada uno de los torques de entrada del modelo. En el caso en que el torque se aplica a  $\tau_1$ , se nota como se mantiene una respuesta oscilatoria en ambos ángulos, pero el ángulo de  $\theta_1$  se mantiene positivo. De manera concreta, este fenómeno implicaría que el primer segmento de brazo comenzaría a oscilar entre dos ángulos dada una entrada de este tipo. Mientras tanto, al aplicar un torque a  $\tau_2$  se nota como el ángulo  $\theta_2$  continua aumentando. En este caso, esto implicaría que el segundo segmento del brazo estaría dando vueltas en dirección horaria de manera continua.

En las figuras 3.10 y 3.11 se muestran las respuestas ante una entrada de impulso en cada una de las entradas, dada una condición inicial cero. Esta prueba nos permite caracterizar la respuesta ante una entrada repentina si el sistema se encuentra en un estado inicial estable en cero. En ambos casos, se nota como el segmento de brazo correspondiente al torque de entrada particular cambia de signo. En el caso de  $\theta_1$ , este empieza a oscilar en ángulos mayores a cero cuando se aplica un impulso a  $\tau_1$ . Similarmente,  $\theta_2$  empieza a oscilar en ángulos mayores a cero cuando se aplica el mismo impulso pero a  $\tau_2$ . Lógicamente lo que esto implica es que un impulso aplicado a estas entradas generara una respuesta similar a la respuesta obtenida sin entradas, pero esta sera desplazada cierta cantidad de grados. De nuevo analizada con el concepto de un péndulo doble, si se empuja el segmento interior, induciendo un torque, este segmento empezara a aumentar su ángulo debido a la fuerza aplicada, hasta que empiece a oscilar en un punto debido a las fuerzas inducidas por el segundo segmento.

## Capítulo 5

# Conclusiones y recomendaciones

### 5.1. Conclusiones

El modelo que se simula provee un interesante caso de estudio para analizar. Al ser un sistema no lineal, las respuestas que provee pueden ser complicadas y difíciles de analizar. De ahí la razón que en el trabajo que se utilizó como punto de partida (GUECHI y col. 2018) se linealiza el sistema antes de intentar controlarlo. Aun así, únicamente usando el sistema no lineal se logró llegar a varias conclusiones sobre las características del mismo. Se noto que la respuesta natural del sistema era oscilatoria. Este factor es posiblemente el más importante, ya que como se observó en el trabajo de GUECHI y col. 2018, reducir esta respuesta oscilatoria para lograr controlar el brazo fue el propósito de todo el trabajo. Se noto a través de las diferentes pruebas y simulaciones que la reducción en las oscilaciones de los ángulos de salida se podían lograr a través de un control cuidadoso de los torques de entrada y las condiciones iniciales. Dadas las diferentes simulaciones que se hicieron, se podría decir que el brazo podría controlarse sin linearizar aplicando una señal de entrada cuidadosamente diseñada. Si se utilizaran pulsaciones correspondientes a la oscilación del sistema, podría en teoría lograrse que el brazo se mantenga en una posición fija. El problema con esto probablemente sería que ese tipo de pulsaciones variarían de manera importante dependiendo de la posición actual de cada segmento del brazo, además de que aplicar estos torques de manera instantánea utilizando un motor o algo similar sería difícil dado el tiempo de respuesta que pueden tener. Un factor importante que no se considera en el modelo es la presencia de fricción en las articulaciones del brazo. Esta fricción serviría para atenuar la respuesta oscilatoria, pero también implicaría un cambio en cuanto torque se debe ejercer para lograr una respuesta estable.

### 5.2. Recomendaciones

- Considerar el factor de fricción en las articulaciones del brazo
- Considerar los tiempos de respuesta y curvas de torque de los motores eléctricos que producirían los torques de entrada
- Linearizar el modelo a utilizar para generar esquemas de control más apropiados

# Bibliografía

- Departamento de Trabajo de los Estados Unidos (mayo de 2019). *Industrial Robots and Robot System Safety*.  
URL: [https://www.osha.gov/dts/osta/otm/otm\\_iv/otm\\_iv\\_4.html#1](https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html#1).
- GUECHI, El-Hadi y col. (2018). *Model predictive control of a two-link robot arm*.
- Semanario Universidad (2019). “Arcos lab: Con la mirada puesta en el futuro”. En: *Ciencia y Tecnología*.
- Sivraj, Kadirimangalam Jahnavi; P. (2017). “Teaching and Learning Robotic Arm Model”. En: *Teaching and Learning Robotic Arm Model*. IEEE.

## Apéndice A

# Código de s-function

```
1 function sfun_mve(block)
2 % Plantilla para implementar modelos en variable de estado sencillos
3 % como una S-Function
4 %
5 % El nombre del archivo y el de la funcion (ver linea 1) deben ser
   iguales
6
7 %
8 % Aca solo se llama a la funcion setup, no agregue nada mas a esta
   funcion
9 % principal
10 %
11 setup(block);
12
13 % -----
14 % Function: setup =====
15 % Abstract:
16 % Aca se definen las caracterisiticas basicas del bloque de la S-
   function:
17 %   - Puertos de entrada
18 %   - Puertos de salida
19 %   - Definicion de parametros
20 %   - Opciones
21 %
22 %
23 function setup(block)
24
25 % Se registra el numero de puertos de entrada y salida
26 block.NumInputPorts = 2; % Cantidad de entradas
27 block.NumOutputPorts = 2; % Cantidad de salidas
28
```

```

29 % Setup port properties to be inherited or dynamic
30 block.SetPreCompInpPortInfoToDynamic;
31 block.SetPreCompOutPortInfoToDynamic;
32
33 % Propiedades de los puertos de entrada
34 block.InputPort(1).Dimensions = 1;
35 block.InputPort(1).DatatypeID = 0; % double
36 block.InputPort(1).Complexity = 'Real';
37 block.InputPort(1).DirectFeedthrough = true;
38
39 % Propiedades de los puertos de entrada
40 block.InputPort(2).Dimensions = 1;
41 block.InputPort(2).DatatypeID = 0; % double
42 block.InputPort(2).Complexity = 'Real';
43 block.InputPort(2).DirectFeedthrough = true;
44
45 % Propiedades de los puertos de salida
46 block.OutputPort(1).Dimensions = 1;
47 block.OutputPort(1).DatatypeID = 0; % double
48 block.OutputPort(1).Complexity = 'Real';
49 block.OutputPort(1).SamplingMode = 'Sample';
50
51 block.OutputPort(2).Dimensions = 1;
52 block.OutputPort(2).DatatypeID = 0; % double
53 block.OutputPort(2).Complexity = 'Real';
54 block.OutputPort(2).SamplingMode = 'Sample';
55
56 % Numero de parametros
57 block.NumDialogPrms = 2; % en este caso hay dos parametros de
    entrada
58
59 % Tiempo de muestreo
60 block.SampleTimes = [0, 0]; % Tiempo de muestreo continuo
61 % Definicion de la cantidad de variables de estado
62 block.NumContStates = 4; % En este caso, hay 4 variables de estado
63
64 block.SimStateCompliance = 'DefaultSimState';
65 % -----
66 % Ahora se registran los metodos internos de la S-function
67 % -----
68 block.RegBlockMethod('InitializeConditions', @Inicializacion);
69 block.RegBlockMethod('Outputs', @Salidas);
70 block.RegBlockMethod('Derivatives', @ModeloEstados);
71 block.RegBlockMethod('SetInputPortSamplingMode',
    @SetInputPortSamplingMode); % Necesario para tener dos salidas
72 %end setup

```



```

73
74 function Inicializacion(block)
75
76 block.ContStates.Data = block.DialogPrm(1).Data; % Define los estados
       iniciales con el primer vector de parametros
77 %end Inicializacion
78
79 %
80 function Salidas(block)
81 % Aqui se escriben las ecuaciones de salida
82 x = block.ContStates.Data; % el estado actual
83 block.OutputPort(1).Data = rad2deg((x(1))); % En este caso, la salida 1
       es el angulo 1 en grados
84 block.OutputPort(2).Data = rad2deg((x(2))); % En este caso, la salida 2
       es el angulo 2 en grados
85 %end Salidas
86
87 function ModeloEstados(block)
88 % Aqui se escribe la funcion que calcula las derivadas de las variables
       de
89 % estado
90 x = block.ContStates.Data; % el valor del estado actual
91 theta1 = x(1);
92 theta2 = x(2);
93 omega1 = x(3);
94 omega2 = x(4);
95 tau1 = block.InputPort(1).Data; % el valor del torque 1 actual
96 tau2 = block.InputPort(2).Data; % el valor del torque 2 actual
97 m1 = block.DialogPrm(2).Data(1); % el valor recibido de m1
98 m2 = block.DialogPrm(2).Data(2); % el valor recibido de m2
99 l1 = block.DialogPrm(2).Data(3); % el valor recibido de l1
100 l2 = block.DialogPrm(2).Data(4); % el valor recibido de l2
101 g = 9.81;
102
103 c1 = -(m2*l1*l2*(2*omega1*omega2 + (omega1^2))*sin(theta2));
104 c2 = -(m2*l1*l2*omega1*omega2*sin(theta2));
105
106 g1 = -((m1+m2)*g*l1*sin(theta1) + m2*g*l2*sin(theta1+theta2));
107 g2 = -(m2*g*l2*sin(theta1+theta2));
108
109 r1 = tau1 - c1 - g1;
110 r2 = tau2 - c2 - g2;
111
112 d1 = ((m1+m2)*(l1^2)) + (m2*(l2^2)) + (2*m2*l1*l2*cos(theta2));
113 d2 = (m2*(l2^2)) + (m2*l1*l2*cos(theta2));
114 d3 = d2;

```

```

115 d4 = m2*(l2 ^ 2);
116 det = (d1*d4 - d2*d3);
117
118 omega1dot = ((d4 * r1) - (d2 * r2)) / (det);
119 omega2dot = ((d1 * r2) - (d3 * r1)) / (det);
120
121 theta1dot = omega1;
122 theta2dot = omega2;
123
124 block.Derivatives.Data = [theta1dot; theta2dot; omega1dot; omega2dot];
    % actualizacion del bloque de la S-function
125 %end ModeloEstados
126
127 function SetInputPortSamplingMode(s, port, mode)
128 s.InputPort(port).SamplingMode = mode;
129 %end SetInputPortSamplingMode

```